

CS 442 / 542

- Compiling Arrays

Compiling Arrays

```
int num1;  
int num2;  
int i;  
int num3[11];  
num1 = 10 + 2;  
num2 = 20 * 3 + num1;  
i = 7;  
num3[i] = 500;  
num1 = num2 + num3[i] * 10;  
print num1;  
print num2;  
print num3[i];
```

Compiling Arrays

- Add a production to the grammar for array declarations
- Add a production so the left hand side of an assignment statement can be an array element reference
- Add a production so an array element reference can be used in an expression

Compiling Arrays

- Keep an attribute in the symbol table that indicates the size of the array.
- Use this information to generate a directive to allocate space for the array
- This works for global arrays

```
num3: .space    44
```

Compiling Arrays

Example Assembler Code for Array Assignment

Example Assignment statement: `num3[i] = 500;`

```
li    $t2, 500
lw    $t1, i
la    $t3, num3
sll   $t1, $t1, 2
add   $t3, $t3, $t1
sw    $t2, 0($t3)
```

Compiling Arrays

Example Assembler Code using an array element in an expression

Example statement: `num1 = num2+num3[i] * 10;`

```
lw    $t4, num2
lw    $t5, i
la    $t6, num3
sll   $t5, $t5, 2
add   $t6, $t6, $t5
lw    $t6, 0($t6)
li    $t7, 10
mul   $t8, $t6, $t7
add   $t6, $t4, $t8
sw    $t6, num1
```

Compiling Arrays: Bubble Sort

```
i = 0;
while ( i < 10) {
    j = 0;
    while (j < 10-i) {
        if (num3[j] > num3[j+1]) {
            temp = num3[j];
            num3[j] = num3[j+1];
            num3[j+1] = temp;
        }
        j= j+1;
    }
    i = i+1;
}
```

Compiling Arrays: Bubble Sort

L3:

```
li    $t0, 0  
sw    $t0, i
```

L9:

```
lw    $t0, i  
li    $t1, 10  
bge   $t0, $t1, L5  
li    $t0, 0  
sw    $t0, j
```

L8:

```
lw    $t0, j  
li    $t1, 10  
lw    $t2, i  
sub   $t3, $t1, $t2  
bge   $t0, $t3, L6
```


Compiling Arrays: Bubble Sort

```
lw    $t0, j
la    $t1, num3
sll   $t0, $t0, 2
add   $t1, $t1, $t0
lw    $t1, 0($t1)
lw    $t0, j
li    $t2, 1
add   $t3, $t0, $t2
la    $t0, num3
sll   $t3, $t3, 2
add   $t0, $t0, $t3
lw    $t0, 0($t0)
ble   $t1, $t0, L7
```

Compiling Arrays: Bubble Sort

```
lw    $t0, j
la    $t1, num3
sll   $t0, $t0, 2
add   $t1, $t1, $t0
lw    $t1, 0($t1)
sw    $t1, temp
lw    $t0, j
lw    $t1, j
li    $t2, 1
add   $t3, $t1, $t2
la    $t1, num3
sll   $t3, $t3, 2
add   $t1, $t1, $t3
lw    $t1, 0($t1)
la    $t2, num3
sll   $t0, $t0, 2
add   $t2, $t2, $t0
sw    $t1, 0($t2)
```

Compiling Arrays: Bubble Sort

```
lw    $t0, j
li    $t1, 1
add   $t2, $t0, $t1
lw    $t0, temp
la    $t1, num3
sll   $t2, $t2, 2
add   $t1, $t1, $t2
sw    $t0, 0($t1)
```

L7:

```
lw    $t0, j
li    $t1, 1
add   $t2, $t0, $t1
sw    $t2, j
b     L8
```

L6:

```
lw    $t0, i
li    $t1, 1
add   $t2, $t0, $t1
sw    $t2, i
b     L9
```

L5: